# Physically-Based Models with Rigid and Deformable Components

**Demetri Terzopoulos**
**Andrew Witkin**

**Schlumberger Palo Alto Research**
**3340 Hillview Avenue, Palo Alto, CA 94304**

**Abstract**

*In prior work we proposed a class of physically-based models suitable for animating flexible objects in simulated physical environments [1]. Our original formulation works well in practice for models whose shapes are moderately to highly deformable, but it tends to become numerically ill-conditioned as we increase the rigidity of the models. The present paper develops an alternative formulation of deformable models. We decompose deformations into a reference component, which may represent an arbitrary shape, and a displacement component allowing deformation away from this reference shape. The reference component evolves according to the laws of rigid-body dynamics. Equations of nonrigid motion based on linear elasticity govern the dynamics of the displacement component. With nonrigid and rigid dynamics operating in unison, this hybrid formulation yields well-conditioned discrete equations, even for complicated refrence shapes, particularly as the rigidity of models is increased beyond the stability limits of our prior formulation. We illustrate the application of our deformable models to a physically-based computer animation project.*

**Keywords:** Modeling, Animation, Deformation, Elasticity, Dynamics, Simulation

## 1. Introduction

Conventional computer animation is kinematic. The animation of graphics objects often requires the coordinated motion of multiple geometric primitives each involving multiple variables such as position, orientation, scale, etc. To synthesize convincing motions, the animator must specify the variables at each instant in time while satisfying kinematic constraints. A standard scheme for rendering the task less onerous is to automatically spline trajectories through keyframes. Often the results are not entirely satisfactory—motions, especially as they increase in complexity, tend to inherit unnatural qualities. In short, creating natural-looking animation kinematically requires patience and expertise [2].

An alternative to kinematic animation is dynamic animation. The latter offers unsurpassed realism because it makes use of fundamental physical principles. Even inexperienced users can create realistic motions by applying forces to dynamic models in simulated physical worlds. Numerical procedures automatically generate time-varying values for variables in accordance with the laws of Newtonian mechanics. Like real-world objects (and quite unlike conventional, purely geometric models) dynamic models have an active, natural response to applied forces. Such models encourage computer animators to think more like stage choreographers. Choreographers need not concern themselves with every kinematic detail of a routine—physics inevitably dictates the low-level motions of dancers—but concentrate instead on more abstract qualities such as timing, rhythm, and style. By incorporating physics into computer animation, we are able to choreograph motions in similar terms by controlling the simulation through physical parameters, initial conditions, and applied forces. It is therefore not surprising to see a growing interest in the development of physically-based models for the purposes of dynamic animation [3–9].

### 1.1. Physical Simulation of Nonrigid Objects

Physical simulation is especially indispensable when animating continuously flexible objects. In [1], we propose a class of physically-based models that describe the shapes and motions of deformable curve, surface, and solid primitives. These primitives simulate elastic materials such as string, rubber, cloth, paper, sheet metal, or sponge. Our results demonstrate complex motions arising from the interaction of deformable models with ambient media and impenetrable obstacles. Any attempt to recreate the realism of these free-form motions kinematically—that is, without making use of the physical principles underlying the dynamics of nonrigid bodies—seems contrived and unreasonably tedious.

The deformable models in [1] are based on elasticity theory [10]. The (Lagrangian) equations of nonrigid motion are expressed in terms of position functions in Euclidean 3-space. These functions are parametric in the material (intrinsic) coordinates of the model—they directly locate each of its points in space as a function of time. The partial differential equations of motion include a nonlinear elastic force associated with the deformable body. We designed this force to be invariant to rigid-body motion, which imparts no deformation. Nonlinearity results

because the elastic force attempts to restore the shape of the deformed body to a prescribed, undeformed or rest shape that is generally nontrivial. The shape is defined by as many nonvanishing fundamental tensors as may be necessary to specify a reference shape up to a rigid-body transformation (e.g., for a deformable curve, the required tensors reduce to the familiar arc-length, curvature, and torsion functions along the prescribed undeformed curve).

The advantage of nonlinear elasticity, on the one hand, is that it is in principle the most accurate way to characterize the behavior of certain elastic phenomena, such as the large deformation of shells. On the other hand, the nonlinear formulation can lead to serious practical difficulties in the numerical implementation of deformable models. First the discrete equations involved become increasingly ill-conditioned as one tries to increase the rigidity of the model or as the rest shapes are made more complex. Second, relatively complicated algorithms are needed to integrate nonlinear, time-varying partial differential equations and to deal with the probable nonuniqueness of their solutions. The computational cost of robust solution methods tends to be high.

### 1.2. Decomposition into Reference and Displacement Components

Since linear elasticity theory circumvents the complexities of its nonlinear counterpart, we would like to exploit it in the formulation of deformable models for computer graphics. Now, the nonlinear formulation in [1] reduces to a linear model when the rest shape has trivially zero fundamental tensors; i.e., collapsed to a point. Such a restriction is clearly unreasonable. Another possibility, which we have attempted with limited success, is to linearize the equations by extracting the nonlinearity from the elastic force to approximate its effect as an explicit, external force. Unfortunately, the explicit forces tend to degrade the stability of our time integration algorithms.

In this paper, we define deformable models for computer graphics applications which enjoy the benefits of linear elasticity. Rather than being based explicitly on a single set of position functions as in [1], the new model is based on two sets of dependent functions: functions that determine a reference configuration for the body in 3-space, and functions that determine the displacements of material points away from the reference configuration. Clearly, the 3-space positions of points can be determined by adding the displacement component to the reference component.

The elastic behavior of the deformable model manifests itself only in the displacement component, which defines the deformation mode of the model. The deformation mode is governed by linear elasticity, where zero displacement can imply an arbitrary shape determined by the reference configuration component. But the reference component represents a fixed set of reference positions in 3-space, hence the position and attitude of the rest shape will be fixed. In order for the deformable model to permit a rigid-body motion mode in addition to an elastic mode, we allow the reference component to evolve over time according to the laws of rigid-body dynamics [11]. Thus, we obtain a *hybrid* model which includes both rigid and deformable dynamics. With regard to numerical implementation, the hybrid formulation of deformable models can offer an important benefit—it leads to discrete equations that remain well-conditioned as we make the model more rigid.

The remainder of this paper is organized as follows: Section 2 describes the geometric representation underlying the hybrid formulation. Section 3 develops the equations of motion governing the hybrid model. Section 4 develops the energy of linear elastic deformation. Section 5 describes our numerical solution. Section 6 presents an application of our deformable models to a physically-based animation project.

## 2. Geometric Representation

Let $\mathrm{u}$ be the intrinsic or material coordinates of points in a body $\Omega$. For a solid body $\mathrm{u} = (u_1, u_2, u_3)$ has three coordinates. For a surface $\mathrm{u} = (u_1, u_2)$ and for a curve $\mathrm{u} = (u_1)$. In the three cases, respectively, and without loss of generality, $\Omega$ will be the unit interval $[0,1]$, the unit square $[0,1]^2$, and the unit cube $[0,1]^3$.

The positions of points in the body relative to an inertial frame of reference $\Phi$ in Euclidean 3-space are given by a time-varying vector valued function of the material coordinates

$$\mathbf{x}(\mathrm{u},t) = [x_1(\mathrm{u},t), x_2(\mathrm{u},t), x_3(\mathrm{u},t)]', \tag{1}$$

where the prime denotes the transpose operator. We indicate a 3-space vector in bold face, while its elements are written in italic face.

We represent a deformable body as the sum of a reference component

$$\mathbf{r}(\mathrm{u}) = [r_1(\mathrm{u}), r_2(\mathrm{u}), r_3(\mathrm{u})]' \tag{2}$$

and an elastic component

$$\mathbf{e}(\mathrm{u},t) = [e_1(\mathrm{u},t), e_2(\mathrm{u},t), e_3(\mathrm{u},t)]'. \tag{3}$$

It is convenient to express both components in body coordinates; that is, relative to a reference frame $\phi$ (Fig. 1) whose origin coincides with the body's center of mass

$$\mathbf{c}(t) = \int_\Omega \mu(\mathrm{u})\mathbf{x}(\mathrm{u},t)\,d\mathrm{u}, \tag{4}$$

where $\mu(\mathrm{u})$ is the mass density of the deformable body ($\Omega$ is assumed to be the domain of integration for integrals with respect to $\mathrm{u}$ and is henceforth suppressed). We denote the positions of mass elements in the body relative to $\phi$ by

$$\mathbf{q}(\mathrm{u},t) = \mathbf{r}(\mathrm{u}) + \mathbf{e}(\mathrm{u},t). \tag{5}$$

The body frame $\phi$ translates and rotates along with the deformable body. The positions of mass elements relative to the inertial frame $\Phi$ is given by $\mathbf{x}(\mathrm{u},t)$ as shown in the figure. To obtain $\mathbf{x}$, we need to know the orientation $\boldsymbol{\theta}(t)$ of $\phi$ relative to $\Phi$ and the displacement $\mathbf{c}$ between the two frames. The linear and angular velocity of $\phi$ relative to $\Phi$ are

$$\mathbf{v}(t) = \dot{\mathbf{c}}(t); \qquad \boldsymbol{\omega}(t) = \dot{\boldsymbol{\theta}}(t), \tag{6}$$

and the velocity of mass elements relative to $\Phi$ is

$$\dot{\mathbf{x}}(\mathrm{u},t) = \mathbf{v}(t) + \boldsymbol{\omega}(t) \times \mathbf{q}(\mathrm{u},t) + \dot{\mathbf{e}}(\mathrm{u},t). \tag{7}$$

An overstruck dot denotes a time derivative $d/dt$ or $\partial/\partial t$ as appropriate.

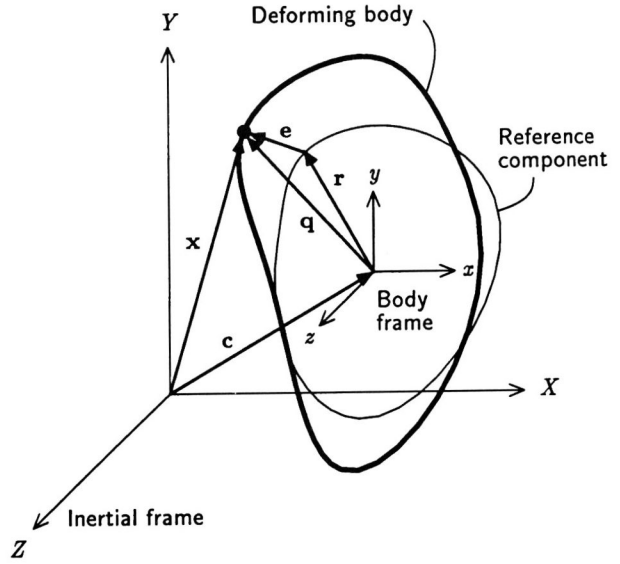


**Figure 1.** Geometric representation. Shape is decomposed into reference and deformation components.

## 3. Equations of Motion

A deformable model is described completely by the positions $\mathbf{x}(\mathrm{u},t)$, velocities $\dot{\mathbf{x}}(\mathrm{u},t)$, and accelerations $\ddot{\mathbf{x}}(\mathrm{u},t)$ of its mass elements as a function of material coordinates u and time $t$. When these functions are expressed in the inertial frame $\Phi$ directly from (1), without making reference to a body frame $\phi$ as in (7), the Lagrange equation of motion governing $\mathbf{x}(\mathrm{u},t)$ takes on a relatively simple form [1]:

$$\mu\ddot{\mathbf{x}} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{x}}\mathcal{E} = \mathbf{f}, \tag{8}$$

where $\mu(\mathrm{u})$ is the mass density, $\gamma(\mathrm{u})$ is the damping density (here a scalar, but generally a matrix), and $\mathbf{f}(\mathbf{x},t)$ represents the net external forces. This is a partial differential equation (due to the dependence of $\delta_{\mathbf{x}}\mathcal{E}$ on $\mathbf{x}$ and its partial derivatives with respect to u—see below). Given appropriate conditions for $\mathbf{x}$ on the boundary of $\Omega$ and initial conditions $\mathbf{x}(\mathrm{u},0)$, $\dot{\mathbf{x}}(\mathrm{u},0)$, we have a well-posed initial-boundary-value problem (second-order in time and of the hyperbolic-parabolic type).

The external forces $\mathbf{f}$ are dynamically balanced against the force terms intrinsic to the deformable model, which are found on the left hand side of (8). The first term is the inertial force due to the model's distributed mass as it resists acceleration. The second term is a velocity dependent (viscous) damping force which dissipates the kinetic energy of the body's mass elements as they move through a viscous ambient medium. The third term is the elastic force due to the deformation of the model away from its natural reference shape.

The elastic force is conveniently expressed as $\delta_{\mathbf{x}}\mathcal{E}$, a variational derivative [12] of a deformation energy $\mathcal{E}(\mathbf{x})$ associated with the model. The nonnegative functional $\mathcal{E}$ measures the potential energy associated with an instantaneous elastic deformation of the body. Its value increases monotonically with the magnitude of the deformation.

Equation (8) is used in [1], where it proves viable for models with moderately high flexibilities. However, our experiments with increasingly rigid models indicate a rapid deterioration of the numerical conditioning of the associated discrete equations. This is due to the fact that in order to increase rigidity using this formulation, we must increase the values of parameters that make $\mathcal{E}$ more nonquadratic and, consequently, make (8) more nonlinear.

The numerical degeneration may be avoided by decomposing $\mathbf{x}$ into the reference component $\mathbf{r}$ and deformation component $\mathbf{e}$ as in (7), and reformulating the equation of motion to treat rigid-body motion explicitly, hence permitting us to use a purely quadratic elastic functional $\mathcal{E}$. The numerical conditioning of the new formulation will tend to improve as the model becomes more rigid, tending in the limit to very well-conditioned, rigid-body dynamics. The capability of modeling nonrigid bodies is retained, but attempting extremely nonrigid models like stretchy rubber sheets with the hybrid formulation may yield unrealistic results, due to the simple connection of the deformation to a rigid reference shape through linear elastic forces.

To obtain the equations of motion for the the unknown functions $\mathbf{v}$, $\boldsymbol{\omega}$, and $\mathbf{e}$ under the action of an applied forces $\mathbf{f}$, we transform the kinetic energy which governs the deformable body using Lagrangian mechanics. Assuming small deformations, this yields three coupled, differential equations:

$$m\dot{\mathbf{v}} + \frac{d}{dt}\int \mu\dot{\mathbf{e}}\,d\mathrm{u} + \int \gamma\dot{\mathbf{x}}\,d\mathrm{u} = \mathbf{f}^{\mathbf{v}}, \tag{9a}$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) + \frac{d}{dt}\int \mu\mathbf{q}\times\dot{\mathbf{e}}\,d\mathrm{u} + \int \gamma\mathbf{q}\times\dot{\mathbf{x}}\,d\mathrm{u} = \mathbf{f}^{\boldsymbol{\omega}}, \tag{9b}$$

$$\mu\ddot{\mathbf{e}} + \mu\dot{\mathbf{v}} + \mu\boldsymbol{\omega}\times(\boldsymbol{\omega}\times\mathbf{q}) + 2\mu\boldsymbol{\omega}\times\dot{\mathbf{e}} + \mu\dot{\boldsymbol{\omega}}\times\mathbf{q} + \gamma\dot{\mathbf{x}} + \delta_{\mathbf{e}}\mathcal{E} = \mathbf{f}. \tag{9c}$$

Here, $m = \int \mu\,d\mathrm{u}$ is the total mass of the body, and the inertia tensor $\mathbf{I}$ is a $3\times 3$ symmetric matrix with entries

$$I_{ij} = \int \mu(\delta_{ij}\mathbf{q}^2 - q_i q_j)\,d\mathrm{u}, \tag{10}$$

where $\mathbf{q} = [q_1, q_2, q_3]'$ and $\delta_{ij}$ is the Kronecker delta. The applied force $\mathbf{f}(\mathrm{u},t)$ contributes to elastic deformation, as well as to a net translational force $\mathbf{f}^{\mathbf{v}}(t)$ and net torque $\mathbf{f}^{\boldsymbol{\omega}}(t)$ acting on the center of mass:

$$\mathbf{f}^{\mathbf{v}} = \int \mathbf{f}\,d\mathrm{u}; \qquad \mathbf{f}^{\boldsymbol{\omega}} = \int \mathbf{q}\times\mathbf{f}\,d\mathrm{u}. \tag{11}$$

We derive this system of equations in Appendix A.

Let us examine equations (9) in detail. Equations (9a) and (9b) describe $\mathbf{v}$ and $\boldsymbol{\omega}$, the translational and rotational motion of the body's center of mass. Together, these ordinary differential equations describe the motion of the body frame $\phi$ relative to the inertial frame $\Phi$. The partial differential equation (9c) describes, relative to $\phi$, the deformation $\mathbf{e}$ of the model from its reference shape $\mathbf{r}$.

The first two terms on the left hand side of (9a) represent the total inertial forces experienced by the center of mass. The first is due to the total moving mass of the body as if it were concentrated at $\mathbf{c}$, while the second is due to the total displacement motion of mass elements about the reference component $\mathbf{r}$. The third term is the total damping force of the moving mass elements. An analogous interpretation in terms of inertial torques holds for

(9b). The first two terms are the inertial torques due to the body's moment of inertia about $\mathbf{c}$ and the total angular momentum due to the displacement motion of mass elements, while the third term is the total damping torque of the elements.

Equation (9c) indicates several inertial forces experienced by individual mass elements as they deform in the body frame $\Phi$. The first term is the simple inertial force of a mass element. The second term is the inertial force due to the linear acceleration of the center of mass. The next three terms are the centrifugal force on mass elements due to the rotation of $\phi$, the Coriolis force due the velocity of the mass elements in $\phi$, and the transverse force on these elements due to the angular acceleration of $\phi$. The penultimate term is the damping force on individual mass elements. The final term, the elastic force due to the deformation of elements away from the reference component, is examined in the next section.

## 4. Elastic Deformation

The elastic force due to deformational displacement $\mathbf{e}(\mathrm{u},t)$ away from the reference component $\mathbf{r}(\mathrm{u})$ is represented in (9c) by $\delta_{\mathbf{e}}\mathcal{E}$, a variational derivative with respect to $\mathbf{e}$ of a elastic potential energy functional $\mathcal{E}$. The general form of $\mathcal{E}$ is

$$\mathcal{E}(\mathbf{e}) = \int E(\mathrm{u}, \mathbf{e}, \mathbf{e}_{\mathrm{u}}, \mathbf{e}_{\mathrm{uu}}, \ldots)\, d\mathrm{u}, \tag{12}$$

an integral over material coordinates of an elastic energy density $E$, which depends on $\mathbf{e}$ and its partial derivatives with respect to material coordinates.

In [1] the elastic functional for a solid deformable model was of the form $\mathcal{E}(\mathbf{x}) = \int |\mathbf{G} - \mathbf{G}^0|^2\, d\mathrm{u}$, a squared normed difference between the first-order or metric tensors (matrices) $\mathbf{G}(\mathbf{x})$ of the deformed body and $\mathbf{G}^0$ of the undeformed body. Elastic functionals for surface and curve models involve additional squared difference terms of second- and third-order tensors. The collection of tensors associated with the undeformed body describe its shape up to rigid-body motions, and $\mathcal{E}$ quantifies the model's actual deformation away from this rigid shape. Thus, the reference component is incorporated into the energy functional which is invariant with respect to rigid-body motion. Such invariance is necessary in the simple equation of motion (8).

The virtue of the new equations of motion (9) is that they make fewer demands on $\mathcal{E}$. Because rigid motion is represented explicitly, $\mathcal{E}$ no longer need be invariant with respect to such motion. All that is required is that $\mathcal{E} = 0$ when $\mathbf{e} = \mathbf{0}$ and that $\mathcal{E}$ increase monotonically with increasing $\mathbf{e}$, as measured by some reasonable norm.

We have at our disposal a class of controlled-continuity generalized spline kernels [13]. These splines are of the form (12) with the potential energy density defined by

$$E = \frac{1}{2} \sum_{m=0}^{p} \sum_{|j|=m} \frac{m!}{j_1! \ldots j_d!} w_j(\mathrm{u}) \left|\partial_j^m \mathbf{e}\right|^2, \tag{13}$$

where $j = (j_1, \ldots, j_d)$ is a multi-index with $|j| = j_1 + \ldots + j_d$, where $d$ is the material dimensionality of the model ($d = 1$ for curves, $d = 2$ for surfaces, and $d = 3$ for solids), and where the partial derivative operator

$$\partial_j^m = \frac{\partial^m}{\partial u_1^{j_1} \ldots \partial u_d^{j_d}}. \tag{14}$$

Thus, $E$ is a weighted combination of partial derivatives of $\mathbf{e}$ of all orders up to $p$. Generally, the smoothness of the allowable deformation increases with increasing $p$. The weighting functions $w_j(\mathrm{u})$ in (13) control the material properties of the deformable model over the material coordinates.

In the interior of the material domain $\Omega$, the variational derivative of $\mathcal{E}$ with the spline density (13) is

$$\delta_{\mathbf{e}}\mathcal{E} = \sum_{m=0}^{p} (-1)^m \Delta_{w_m}^m \mathbf{e}, \tag{15}$$

where

$$\Delta_{w_m}^m = \sum_{|j|=m} \frac{m!}{j_1! \ldots j_d!} \partial_j^m \left(w_j(\mathrm{u}) \partial_j^m\right) \tag{16}$$

is a spatially-weighted iterated Laplacian operator of order $m$. The operator is modified at the boundary, according to the boundary conditions (see [13]).

## 5. Numerical Solution

The equations of motion (9–11) with (15–16) are continuous in material coordinates and time. To numerically simulate the deformable model, we discretize the equations using finite-element or finite-difference approximation methods [14, 15]. First we discretize with respect to material coordinates to obtain semidiscrete equations of motion. The result is a large system of simultaneous ordinary differential equations.

The second step is to integrate the semidiscrete system through time, thus simulating the dynamics of deformable models. We use a semi-implicit time integration procedure which evolves the elastic displacements and rigid-body dynamics from given initial conditions. In essence, the evolving deformation yields a recursive sequence of (dynamic) equilibrium problems, each requiring solution of a *sparse,* linear system whose dimensionality is proportional to the number of nodes comprising the discrete model.

We can use iterative methods to solve these linear systems, as well as direct matrix factorization methods (such as Choleski) [16]. Due to the linear elastic energy density (13), the system matrix system is constant; hence a direct solution method need factorize it only once at the beginning, then simply resolve the vector $\underline{\mathrm{b}}$ at each time step, thus saving significant computation.

Our implementations to date employ the second-order ($p = 2$) controlled-continuity spline model. Appendix B presents implementation details for the case of surfaces.

## 6. Animation Examples

To create animation we simulate numerically the differential equations of motion. After each time step (or every few time steps) in the simulation, we render the models' state data to create successive frames of the animation.

We have implemented curve, surface, and solid deformable models in three dimensions on Symbolics 3600 series Lisp Machines. These machines provide an excellent prototyping environment, but their processors lack the power to support real-time interaction with discrete surface and solid models of modest size (having more than about 100 nodes).

However, we can easily interact with deformable models in real-time within a "flatland" environment, subjecting them to user controlled forces, gravity, forces due to collisions with obstacles, etc. (see [1] for more details on formulating applied forces). Flatland models are planar deformable curves (displayed as wireframes), capable of rigid-body dynamics and elastodynamics (Fig. 2).

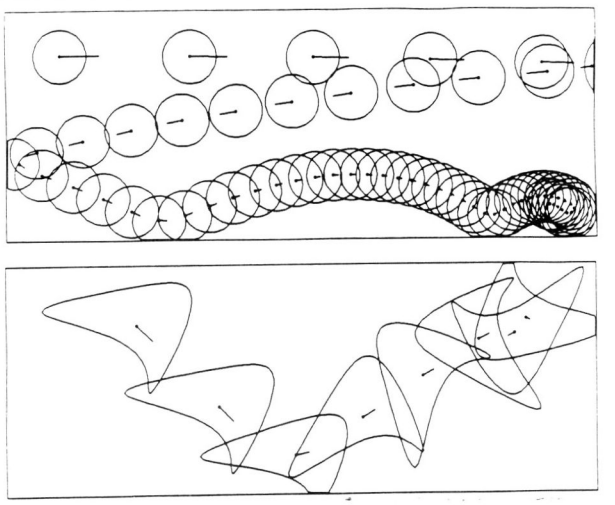

**Figure 2.** Flatland animations. Models are "strobed" while undergoing motion subject to gravity, aerodynamic drag, and collisions against frictionless walls. Velocity vector of the center of mass (dot) is indicated. (top) A deformable hoop. (bottom) An arbitrary shape.

Deformable models offer a powerful alternative to the standard practice of creating animation by key-framing and in-betweening. We have used the formulation in this paper to create the animation *Cooking with Kurt* [17]. The animation was rendered using the modeling testbed system described in [18]. Fig. 3 shows selected still frames. The action begins with live video footage of Kurt Fleischer placing several vegetables on a cutting board. The vegetables "come to life," bouncing and rolling around the kitchen table environment, colliding with one another and with the table-top.

The synthetic vegetables are deformable surface models. Their reference shapes $\mathbf{r}$ were reconstructed from an image of the real vegetables using computer vision techniques, some of which are described in [19–21]. These techniques exploit the "modeling clay" properties of deformable models. They provide principled ways of transforming raw image data into synthetic force fields that sculpt deformable models into shapes consistent with the imaged objects. Simple optimization methods served to bring the synthetic light source, surface albedos, etc. into consistency with the real scene. After the models captured the shapes of the real objects, they were animated in a physically-based table-top environment by simulating the equations of motion. The applied forces include driving forces (jet thrusters), control forces (attitude control gyros), and interaction forces (friction or collisions). The deformable models exhibit deformations, accelerations, collisions, tumbling, and other realistic physical motions. Fig. 3 shows some synthesized motions in progress.

## 7. Conclusion

We proposed novel physically-based models for use in computer graphics animation. Our hybrid deformable models unify rigid-body dynamics with nonrigid-body dynamics. By incorporating a reference component with explicit (six degree-of-freedom) dynamical equations, we are able to exploit a simple linear theory to model free-form elastic deformations. Reduction in computational effort, good conditioning of the numerical equations with increasing rigidity, and the consequent ability to animate flexible objects with complicated natural shapes are among the benefits accrued. Moreover, our hybrid formulation makes it especially convenient to model inelastic deformation such as viscoelasticity and plasticity using dynamic feedback from the displacement component into the reference component which now maintains a shape history. A subsequent paper will describe inelastic models that behave like modeling clay. The hybrid formulation complements our earlier work in elastically deformable models and significantly extends our abilities to create realistic animations of nonrigid objects in simulated physical environments.

### Acknowledgements

We wish to thank our fellow goop scientists: Michael Kass assisted with the derivation in Appendix A. Kurt Fleischer and he played major roles in creating *Cooking with Kurt.* John Platt made valuable contributions to our earlier work on deformable models.

## A. Derivation of the Equations of Motion

We apply Lagrangian mechanics to the kinetic energy that governs deformable models:

$$\mathcal{T} = \frac{1}{2}\int \mu \dot{\mathbf{x}} \cdot \dot{\mathbf{x}}\, du, \tag{17}$$

where $\dot{\mathbf{x}}(u,t)$ is the instantaneous velocity of mass elements. It can be rewritten, using (7), in terms of the geometric representation of Fig. 1 as

$$\mathcal{T} = \int T\, du = \frac{1}{2}\int \mu(\mathbf{v}+\boldsymbol{\omega}\times\mathbf{q}+\dot{\mathbf{e}})\cdot(\mathbf{v}+\boldsymbol{\omega}\times\mathbf{q}+\dot{\mathbf{e}})\, du. \tag{18}$$

Expanding,

$$\mathcal{T} = \sum_{k=1}^{6}\mathcal{T}_k = \sum_{k=1}^{6}\int T_k\, du, \tag{19}$$

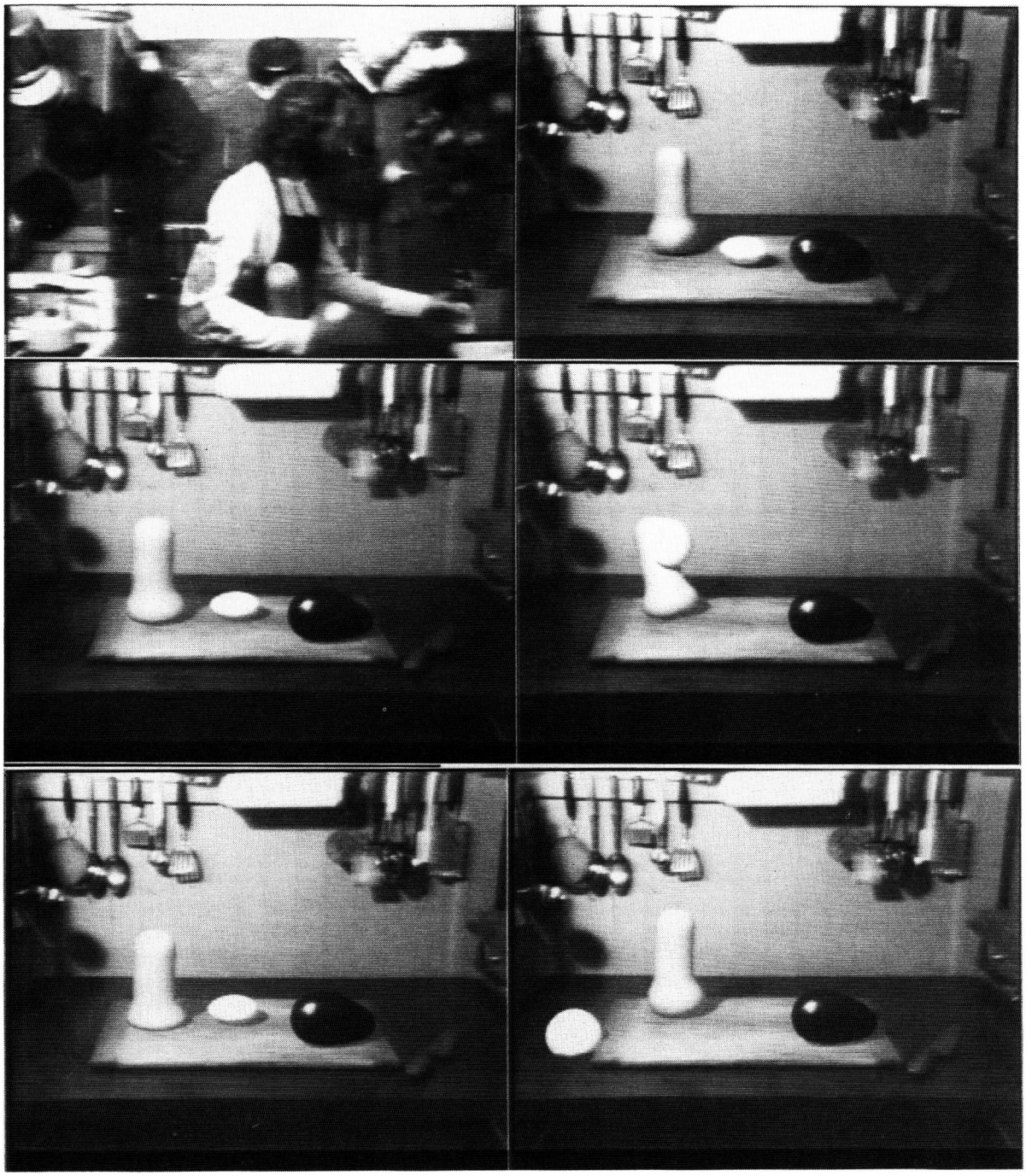

**Figure 3.** Selected still frames from *Cooking with Kurt* [17] showing live action and animation of deformable models in a simulated physical environment. (Top left) Kurt Fleischer with real vegetables. (Top right) Real vegetables on cutting board. (Middle left) Reconstructed deformable-model veggies matted into background scene. (Middle right) Elastic collision. (Bottom left) Bouncing (note deformed base on large gourd). (Bottom right) Rolling.

where the integrands are

$$T_1 = \frac{1}{2}\mu\mathbf{v}\cdot\mathbf{v}, \qquad T_2 = \mu\mathbf{v}\cdot\dot{\mathbf{e}}, \qquad T_3 = \frac{1}{2}\mu\dot{\mathbf{e}}\cdot\dot{\mathbf{e}},$$
$$T_4 = \frac{1}{2}\mu(\boldsymbol{\omega}\times\mathbf{q})\cdot(\boldsymbol{\omega}\times\mathbf{q}), \quad T_5 = \mu\boldsymbol{\omega}\times\mathbf{q}\cdot\dot{\mathbf{e}}, \tag{20}$$
$$T_6 = \mu\mathbf{v}\cdot\boldsymbol{\omega}\times\mathbf{q}.$$

Velocity dependent energy dissipation may be incorporated in terms of the (Raleigh) dissipation functional

$$\mathcal{F} = \int F\,d\mathrm{u} = \frac{1}{2}\int \gamma\dot{\mathbf{x}}\cdot\dot{\mathbf{x}}\,d\mathrm{u}, \tag{21}$$

where $\gamma(\mathrm{u})$ is a damping density. Note that (21) has the same form as (19) with $\gamma$ replacing $\mu$. Hence, we express

$$\mathcal{F} = \sum_{k=1}^{6}\mathcal{F}_k = \sum_{k=1}^{6}\int F_k\,d\mathrm{u}$$

in the representation of Fig. 1, where the associated integrands $F_1$ to $F_6$ are readily obtained from (20) by replacing $\mu$ with $\gamma$.

Using the functionals $\mathcal{T}$ and $\mathcal{F}$ and observing that $\mathcal{E}$ does not depend on $\mathbf{v}$ and $\boldsymbol{\omega}$, the equations of motion can be expressed as

$$\begin{aligned}\delta_{\mathbf{v}}\mathcal{T} + \delta_{\mathbf{c}}\mathcal{F} &= \mathbf{f}^{\mathbf{v}},\\ \delta_{\boldsymbol{\theta}}\mathcal{T} + \delta_{\boldsymbol{\omega}}\mathcal{F} &= \mathbf{f}^{\boldsymbol{\omega}},\\ \delta_{\mathbf{e}}\mathcal{T} + \delta_{\dot{\mathbf{e}}}\mathcal{F} + \delta_{\mathbf{e}}\mathcal{E} &= \mathbf{f},\end{aligned} \tag{22}$$

where the $\delta$ operators denote variational derivatives with respect to the subscripted functions. The generalized forces associated with $\mathbf{v}$, $\boldsymbol{\omega}$, and $\mathbf{e}$ are $\mathbf{f}^{\mathbf{v}}$, $\mathbf{f}^{\boldsymbol{\omega}}$, and $\mathbf{f}$.

In view of the time derivatives contained in the functional terms $\mathcal{T}_k$ and $\mathcal{F}_k$, we have

$$\begin{aligned}\delta_{\mathbf{c}}\mathcal{T} + \delta_{\mathbf{v}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\mathbf{v}} + \frac{\partial\mathcal{F}_k}{\partial\mathbf{v}},\\ \delta_{\boldsymbol{\theta}}\mathcal{T} + \delta_{\boldsymbol{\omega}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\boldsymbol{\omega}} + \frac{\partial\mathcal{F}_k}{\partial\boldsymbol{\omega}},\\ \delta_{\mathbf{e}}\mathcal{T} + \delta_{\dot{\mathbf{e}}}\mathcal{F}_k &= \sum_{k=1}^{6}\frac{\partial}{\partial t}\frac{\partial\mathcal{T}_k}{\partial\dot{\mathbf{e}}} - \frac{\partial\mathcal{T}_k}{\partial\mathbf{e}} + \frac{\partial\mathcal{F}_k}{\partial\dot{\mathbf{e}}}.\end{aligned} \tag{23}$$

Now, $\int\mu\mathbf{q}\,d\mathrm{u} = \mathbf{0}$, since it is simply the center of mass and lies at the origin of the body frame $\phi$; hence $\mathcal{T}_6 = T_6 = 0$. The above sums may be derived term by term:

$$\begin{aligned}\sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\mathbf{v}} &= \frac{d}{dt}(m\mathbf{v}) + \frac{d}{dt}\int\mu\dot{\mathbf{e}}\,d\mathrm{u} + \mathbf{0}+\mathbf{0}+\mathbf{0}+\mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial\mathcal{F}_k}{\partial\mathbf{v}} &= \mathbf{v}\int\gamma\,d\mathrm{u} + \int\gamma\dot{\mathbf{e}}\,d\mathrm{u} + \mathbf{0}+\mathbf{0}+\mathbf{0}+\boldsymbol{\omega}\times\int\gamma\mathbf{q}\,d\mathrm{u},\\ \sum_{k=1}^{6}\frac{d}{dt}\frac{\partial\mathcal{T}_k}{\partial\boldsymbol{\omega}} &= \mathbf{0}+\mathbf{0}+\mathbf{0}+\frac{d}{dt}(\mathbf{L}) - \frac{d}{dt}\int\mu\mathbf{q}\times\dot{\mathbf{e}}\,d\mathrm{u} + \mathbf{0},\\ \sum_{k=1}^{6}\frac{\partial\mathcal{F}_k}{\partial\boldsymbol{\omega}} &= \mathbf{0}+\mathbf{0}+\mathbf{0}+\int\gamma\mathbf{q}\times(\boldsymbol{\omega}\times\mathbf{q})\,d\mathrm{u} - \int\gamma\mathbf{q}\times\dot{\mathbf{e}}\,d\mathrm{u}\\ &\qquad - \mathbf{v}\times\int\gamma\mathbf{q}\,d\mathrm{u},\end{aligned} \tag{24}$$

$$\sum_{k=1}^{6}\frac{\partial}{\partial t}\frac{\partial T_k}{\partial\dot{\mathbf{e}}} = \mathbf{0} + \mu\dot{\mathbf{v}} + \frac{d}{dt}(\mu\dot{\mathbf{e}}) + \mathbf{0} + \mu\big(\dot{\boldsymbol{\omega}}\times\mathbf{q} - \boldsymbol{\omega}\times\dot{\mathbf{e}}\big) + \mathbf{0},$$

$$\sum_{k=1}^{6}\frac{\partial T_k}{\partial\mathbf{e}} = \mathbf{0}+\mathbf{0}+\mathbf{0}+\mu\boldsymbol{\omega}\times\big(\boldsymbol{\omega}\times\mathbf{q}\big) + \mu\boldsymbol{\omega}\times\dot{\mathbf{e}} + \mathbf{0},$$

$$\sum_{k=1}^{6}\frac{\partial F_k}{\partial\dot{\mathbf{e}}} = \mathbf{0} + \gamma\mathbf{v} + \gamma\dot{\mathbf{e}} + \mathbf{0} + \gamma\dot{\boldsymbol{\omega}}\times\mathbf{q} + \mathbf{0},$$

The term $\mathbf{L} = \frac{\partial}{\partial\boldsymbol{\omega}}\int\frac{1}{2}\mu(\boldsymbol{\omega}\times\mathbf{q})\cdot(\boldsymbol{\omega}\times\mathbf{q})\,d\mathrm{u} = \int\mu\mathbf{q}\times(\boldsymbol{\omega}\times\mathbf{q})\,d\mathrm{u}$ is known as the angular momentum of the deformable body as it rotates rigidly about the center of mass. It can be shown that

$$\mathbf{L} = \int\mu(\boldsymbol{\omega}\mathbf{q}\cdot\mathbf{q} - \mathbf{q}\mathbf{q}\cdot\boldsymbol{\omega})\,d\mathrm{u} = \mathbf{I}\boldsymbol{\omega} \tag{25}$$

where $\mathbf{I}$ is the inertia tensor whose components are given in (10).

Inserting (24) and (25) into (23), (22) yields the equations of motion (9).

## B. Implementation Details

To illustrate the implementation, we consider the case of surfaces. Curves (solids) involve a straightforward restriction (extension) of the two-parameter equations developed in this section. Letting $\mathrm{u} = (u_1, u_2) = (u, v)$ be the surface's material coordinates and letting $p = 2$ in (15) and (16) yields the variational derivative

$$\begin{aligned}\delta_{\mathbf{e}}\mathcal{E} =& w_{00}\mathbf{e} - (w_{10}\mathbf{e}_u)_u - (w_{01}\mathbf{e}_v)_v\\ &+ (w_{20}\mathbf{e}_{uu})_{uu} + 2(w_{11}\mathbf{e}_{uv})_{uv} + (w_{02}\mathbf{e}_{vv})_{vv},\end{aligned} \tag{26}$$

where the subscripts denote partial derivatives with respect to material coordinates. The functions $w_j(\mathrm{u})$ locally control the partial derivatives of deformational displacement $\mathbf{e}$ of the model. Specifically, $w_{00}$ penalizes the local magnitude of the deformation, $w_{10}$ and $w_{01}$ penalize its local variations, while $w_{20}$, $w_{11}$, and $w_{02}$ penalize its local curvatures.

### B.1. Semidiscretization

We illustrate the semidiscretization step using standard finite-difference approximations. The unit square domain $\Omega = 0 \le u, v \le 1$ of the surface is discretized as a regular $M \times N$ discrete mesh $\Omega^h$ of nodes. The internode spacings are $h_1 = 1/(M-1)$ and $h_2 = 1/(N-1)$ in the $u$ and $v$ coordinate directions respectively. Nodes are indexed by integers $[m, n]$ where $0 \le m \le M$ and $0 \le n \le N$. We approximate the (continuous) vector functions of $(\mathrm{u}, t)$ in (9–11) by arrays of (continuous-time) vector-valued nodal variables: $\mathbf{r}[m,n] = \mathbf{r}(mh_1, nh_2)$, $\mathbf{e}[m,n](t) = \mathbf{e}(mh_1, nh_2, t)$, and $\mathbf{f}[m,n](t) = \mathbf{f}(mh_1, nh_2, t)$. We will suppress the time dependence notation until the next section where we consider integration through time.

The discrete elastic force requires approximating from the nodal variables $\mathbf{e}[m,n]$ the first and second partial derivatives of $\mathbf{e}$ with respect to material coordinates $u$ and

$v$. We define the forward first difference operators

$$D_{10}^{+}(\mathbf{e})[m,n] = (\mathbf{e}[m+1,n] - \mathbf{e}[m,n])/h_1$$
$$D_{01}^{+}(\mathbf{e})[m,n] = (\mathbf{e}[m,n+1] - \mathbf{e}[m,n])/h_2 \quad (27)$$

and the backward first difference operators

$$D_{10}^{-}(\mathbf{e})[m,n] = (\mathbf{e}[m,n] - \mathbf{e}[m-1,n])/h_1$$
$$D_{01}^{-}(\mathbf{e})[m,n] = (\mathbf{e}[m,n] - \mathbf{e}[m,n-1])/h_2. \quad (28)$$

Using (18–19), the forward and backward cross difference operators are

$$D_{11}^{+}(\mathbf{e})[m,n] = D_{10}^{+}(D_{01}^{+}(\mathbf{e}))[m,n],$$
$$D_{11}^{-}(\mathbf{e})[m,n] = D_{10}^{-}(D_{01}^{-}(\mathbf{e}))[m,n], \quad (29)$$

and the central second difference operators are

$$D_{20}(\mathbf{e})[m,n] = D_{10}^{-}(D_{10}^{+}(\mathbf{e}))[m,n],$$
$$D_{02}(\mathbf{e})[m,n] = D_{01}^{-}(D_{01}^{+}(\mathbf{e}))[m,n]. \quad (30)$$

Using the difference operators, we discretize (26) as follows:

$$\begin{aligned}\delta_{\mathbf{e}}\mathcal{E} \approx & w_{00}\mathbf{e}[m,n] \\ & - D_{10}^{-}(w_{10}D_{10}^{+}\mathbf{e})[m,n] - D_{01}^{-}(w_{01}D_{01}^{+}\mathbf{e})[m,n] \\ & + D_{20}(w_{20}D_{20}\mathbf{e})[m,n] + 2D_{11}^{-}(w_{11}D_{11}^{+}\mathbf{e})[m,n] \\ & + D_{02}(w_{02}D_{02}\mathbf{e})[m,n].\end{aligned} \quad (31)$$

Free (natural) boundary conditions are introduced by nullifying the value of difference operators found inside parentheses in (31). Such conditions are appropriate at the boundaries of $\Omega^h$ where these operators would attempt to access nodal variables $\mathbf{e}[m,n]$ outside the discrete domain. Similarly, fractures are introduced by nullifying the values of any difference operators accessing nodal variables on opposite sides of such discontinuities.

If the nodal variables comprising the grid functions $\mathbf{e}[m,n]$ are collected into an $MN$-dimensional vector $\underline{\mathbf{e}}$, the discrete approximation (31) may be written in the grid vector form $\mathbf{K}\underline{\mathbf{e}}$ where $\mathbf{K}$ is an $MN$-dimensional square matrix. Due to the local nature of the finite-difference discretization, $\mathbf{K}$, known as the stiffness matrix, has the desirable computational properties of sparseness and bandedness.

The discrete mass and damping densities are grid functions $\mu[m,n]$ and $\gamma[m,n]$ respectively. Let $\mathbf{M}$ be the mass matrix, a diagonal $MN$-dimensional square matrix with the $\mu[m,n]$ variables as diagonal components, and let $\mathbf{C}$ be the damping matrix constructed analogously from $\gamma[m,n]$.

Using (31), the equations of motion (9) can be expressed in semidiscrete form by the following system of coupled ordinary differential equations:

$$m\frac{d\mathbf{v}}{dt} = \mathbf{g}^{\mathbf{v}}, \quad (32a)$$

$$\frac{d}{dt}(\mathbf{I}\boldsymbol{\omega}) = \mathbf{g}^{\boldsymbol{\omega}}, \quad (32b)$$

$$\mathbf{M}\ddot{\underline{\mathbf{e}}} + \mathbf{C}\dot{\underline{\mathbf{e}}} + \mathbf{K}\underline{\mathbf{e}} = \underline{\mathbf{g}}^{\mathbf{e}}, \quad (32c)$$

where

$$\mathbf{g}^{\mathbf{v}} = h_1 h_2 \left( \sum_{m,n} \underline{\mathbf{f}} - \frac{d}{dt} \sum_{m,n} \mu\dot{\underline{\mathbf{e}}} - \sum_{m,n} \gamma\dot{\underline{\mathbf{x}}} \right), \quad (33a)$$

$$\mathbf{g}^{\boldsymbol{\omega}} = h_1 h_2 \left( \sum_{m,n} \underline{\mathbf{q}} \times \underline{\mathbf{f}} - \frac{d}{dt} \sum_{m,n} \mu\underline{\mathbf{q}} \times \dot{\underline{\mathbf{e}}} - \sum_{m,n} \gamma\underline{\mathbf{q}} \times \dot{\underline{\mathbf{x}}} \right), \quad (33b)$$

$$\underline{\mathbf{g}}^{\mathbf{e}} = \underline{\mathbf{f}} - \mu\ddot{\underline{\mathbf{e}}} - \mu\dot{\mathbf{v}} - \mu\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \underline{\mathbf{q}}) - 2\mu\boldsymbol{\omega} \times \dot{\underline{\mathbf{e}}} + \mu\dot{\boldsymbol{\omega}} \times \underline{\mathbf{q}}. \quad (33c)$$

Note that the integrals in (9–11) are approximated by sums over nodal variables. Some of the terms in (9) have been brought to the right hand side in order to simplify the final step of the solution process.

### B.2. Numerical Integration Through Time

To simulate the dynamics of our model, we integrate the semidiscrete system (32) through time. Dividing an open-ended interval from $t = 0$ into time steps $\Delta t$, the integration procedure computes a sequence of approximations at times $\Delta t, 2\Delta t, \ldots, t, t + \Delta t, \ldots$. Each time-step requires the solution of two algebraic equations for $\mathbf{v}$ and $\boldsymbol{\omega}$, which describe the rigid motion of the body frame $\phi$, in tandem with a linear algebraic system for the displacement component $\underline{\mathbf{e}}$.

Substituting the discrete time approximation $\dot{\mathbf{v}} \approx (\mathbf{v}_{t+\Delta t} - \mathbf{v}_t)/\Delta t$ into (32a), we obtain the integration procedure

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \Delta t \mathbf{g}_t^{\mathbf{v}}/m \quad (34)$$

for the linear velocity of $\phi$ at the next time instant. Similarly, we obtain from (32b)

$$\boldsymbol{\omega}_{t+\Delta t} = \mathbf{I}_{t+\Delta t}^{-1}(\mathbf{I}_t\boldsymbol{\omega}_t + \Delta t \mathbf{g}_t^{\boldsymbol{\omega}}) \quad (35)$$

for the angular velocity of $\phi$ for the next time instant. At each time step, the body is translated by $\mathbf{d} = \Delta t\mathbf{v}_t$, and rotated by an angle of $\theta = \Delta t|\boldsymbol{\omega}_t|$ about the unit vector $\mathbf{a} = [a_1, a_2, a_3]' = \boldsymbol{\omega}_t/|\boldsymbol{\omega}_t|$ using the matrix $\mathbf{R} =$

$$\begin{pmatrix} a_1a_1\,\mathrm{vers}\,\theta + \cos\theta & a_1a_2\,\mathrm{vers}\,\theta - a_3\sin\theta & a_1a_3\,\mathrm{vers}\,\theta + a_2\sin\theta \\ a_2a_1\,\mathrm{vers}\,\theta + a_3\sin\theta & a_2a_2\,\mathrm{vers}\,\theta + \cos\theta & a_2a_3\,\mathrm{vers}\,\theta - a_1\sin\theta \\ a_3a_1\,\mathrm{vers}\,\theta - a_2\sin\theta & a_3a_2\,\mathrm{vers}\,\theta + a_1\sin\theta & a_3a_3\,\mathrm{vers}\,\theta + \cos\theta \end{pmatrix} \quad (36)$$

where $\mathrm{vers}\,\theta = (1 - \cos\theta)$ denotes the versine of $\theta$. The reference and displacement components are transformed as follows:

$$\mathbf{r}_{t+\Delta t} = \mathbf{R}\mathbf{r}_t + \mathbf{d}; \qquad \mathbf{e}_{t+\Delta t} = \mathbf{R}\mathbf{e}_t. \quad (37)$$

Next, substituting the discrete-time approximations $\ddot{\underline{\mathbf{e}}} \approx (\underline{\mathbf{e}}_{t+\Delta t} - 2\underline{\mathbf{e}}_t + \underline{\mathbf{e}}_{t-\Delta t})/\Delta t^2$ and $\dot{\underline{\mathbf{e}}} \approx (\underline{\mathbf{e}}_{t+\Delta t} - \underline{\mathbf{e}}_{t-\Delta t})/2\Delta t$ into (32c), we obtain the procedure

$$\mathbf{A}\underline{\mathbf{e}}_{t+\Delta t} = \underline{\mathbf{g}}_t, \quad (38)$$

where the constant matrix

$$\mathbf{A} = \mathbf{K} + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right) \quad (39)$$

and the effective force vector

$$\underline{\mathbf{b}}_t = \underline{\mathbf{g}}_t^{\mathbf{e}} + \left(\frac{1}{\Delta t^2}\mathbf{M} + \frac{1}{2\Delta t}\mathbf{C}\right)\underline{\mathbf{e}}_t + \left(\frac{1}{\Delta t}\mathbf{M} - \frac{1}{2\Delta t}\mathbf{C}\right)\tilde{\underline{\mathbf{e}}}_t, \quad (40)$$

with

$$\tilde{\underline{\mathbf{e}}}_t = (\underline{\mathbf{e}}_t - \underline{\mathbf{e}}_{t-\Delta t})/\Delta t. \quad (41)$$

Note that (34–41) specifies a semi-implicit recursive procedure which evolves the rigid-body dynamics and elastic displacements from given initial conditions $\mathbf{v}_0$, $\boldsymbol{\omega}_0$, $\underline{\mathbf{e}}_0$, and $\dot{\underline{\mathbf{e}}}_0$. In particular, the displacement $\underline{\mathbf{e}}$ evolves as a time sequence of static equilibrium problems is solved (each a

sparse, linear system of size proportional to the number of nodes comprising the discrete model).

We have employed iterative methods, such as successive over-relaxation (SOR) or the conjugate gradient method, as well as direct methods, such as Choleski factorization, to solve the sparse linear systems (38) (see [15] or [16]). Since **A** is a sparse matrix (due to (31), each equation will have at most 13 nonzero coefficients), we implement the direct method using an efficient, profile storage scheme. A more detailed description of our linear equation solvers is beyond the scope of this paper. Further computational savings can be had by neglecting some of the interaction terms comprising the right hand sides of (33); for example, the centrifugal force may be neglected unless large spins $\boldsymbol{\omega}$ are expected, while the Coriolis force may be neglected unless significant $\dot{\mathbf{e}}$ is expected.

## References


1. **Terzopoulos, D., Platt, J., Barr, A., and Fleischer, K.,** "Elastically deformable models," *Computer Graphics*, **21**, 4, 1987, (Proc. SIGGRAPH) 205–214.
2. **Lassiter, J.,** "Principles of traditional animation applied to 3D computer animation," *Computer Graphics*, **21**, 4, 1987, (Proc. SIGGRAPH) 35–44.
3. **Armstrong, W.W., and Green, M.,** "The dynamics of articulated rigid bodies for purposes of animation," *Proc. Graphics Interface '85*, Montreal, Canada, 1985, 407–415.
4. **Wilhelms, J., and Barsky, B.A.,** "Using dynamic analysis to animate articulated bodies such as humans and robots," *Proc. Graphics Interface '85*, Montreal, Canada, 1985, 97–104.
5. **Girard, M., and Maciejewski, A.A.,** "Computational modeling for the computer animation of legged figures," *Computer Graphics*, **19**, 3, 1985, (Proc. SIGGRAPH), 263–270.
6. **Feynman, C.R.,** Modeling the Appearance of Cloth, MSc thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 1986.
7. **Hoffmann, C.M., and Hopcroft, J.E.,** "Simulation of physical systems from geometric models," *IEEE Journ. Robotics and Automation*, **RA-3**, 3, 1987, 194–206.
8. **Issacs, P.M., and Cohen, M.F.,** "Controlling dynamic simulation with kinematic constraints, behavior functions, and inverse dynamics," *Computer Graphics*, **21**, 4, 1987, (Proc. SIGGRAPH) 215–224.
9. **Barr, A., Barzel, R., Haumann, D., Kass, M., Platt, J., Terzopoulos, D., and Witkin, A.,** Topics in physically-based modeling, ACM SIGGRAPH '87 Course Notes, Vol. 17, Anaheim, CA, 1987.
10. **Landau, L.D., and Lifshitz, E.M.,** *Theory of Elasticity*, Pergamon Press, London, UK, 1959.
11. **Goldstein, H.,** *Classical Mechanics*, Addison-Wesley, Reading, MA, 1950.
12. **Courant, R., and Hilbert, D.,** *Methods of Mathematical Physics*, Vol. I, Interscience, London, 1953.
13. **Terzopoulos, D.,** "Regularization of inverse visual problems involving discontinuities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, **PAMI-8**, 1986, 413–424.
14. **Lapidus, L., and Pinder, G.F.,** *Numerical Solution of Partial Differential Equations in Science and Engineering*, Wiley, New York, NY, 1982.
15. **Zienkiewicz, O.C.,** *The Finite Element Method; Third edition*, McGraw–Hill, London, 1977.
16. **Dahlquist, G., and Bjorck, A.,** *Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ, 1974.
17. **Fleischer, K., Witkin, A., Kass, M., and Terzopoulos, D.,** "Cooking with Kurt," *An Animated Video*, Schlumberger Palo Alto Research, Palo Alto, CA, 1987.
18. **Fleischer, K., and Witkin, A.,** "A modeling testbed," *These proceedings*, 1988.
19. **Terzopoulos, D., Witkin, A., and Kass, M.,** "Symmetry-seeking models and 3D object reconstruction," *International Journal of Computer Vision*, **1**, 1987, 211-221.
20. **Kass, M., Witkin, A., and Terzopoulos, D.,** "Snakes: Active contour models," *International Journal of Computer Vision*, **1**, 1987, 321–331.
21. **Terzopoulos, D.,** "On matching deformable models to images: Direct and iterative solutions," *Topical Meeting on Machine Vision, Technical Digest Series, Vol. 12., Optical Society of America*, Washington, DC, 1987, 160–167.